

IMPROVED INTERNET TRAFFIC ANALYSIS VIA OPTIMIZED SAMPLING

Sean McPherson and Antonio Ortega

Ming Hsieh Department of Electrical Engineering
University of Southern California, Los Angeles, CA, 90089-2564
{smcphers, aortega}@usc.edu

ABSTRACT

Applications to evaluate Internet quality-of-service and increase network security are essential to maintaining reliability and high performance in computer networks. These applications typically use very accurate, but high cost, hardware measurement systems. Alternate, less expensive software based systems are often impractical for use with analysis applications because they reduce the number and accuracy of measurements using a technique called interrupt coalescence, which can be viewed as a form of sampling. The goal of this paper is to optimize the way interrupt coalescence groups packets into measurements so as to retain as much of the packet timing information as possible. Our optimized solution produces estimates of timing distributions much closer to those obtained using hardware based systems. Further we show that for a real Internet analysis application, periodic signal detection, using measurements generated with our method improved detection times by at least 36%.

Index Terms— Sampling Methods, Optimization Methods, Internet

1. INTRODUCTION

Internet traffic measurement systems are used in numerous network analysis applications such as network tomography, security and quality-of-service measuring tools [1, 2]. In general a measurement system is any system connected to an Internet link that extracts information from packet headers, such as packet size, as the packets pass through the system. The measurement system records this information along with a time-stamp indicating, with some processing delay, when the packet arrived at the measurement system.

Generally speaking there are two categories of measurement systems (although hybrid systems do exist): hardware and software measurement systems. Hardware based systems are specially designed to generate very precise time-stamps for each packet received. Software systems, having other processing requirements, are unable to keep up with the processing burden required to generate time-stamps for individual packets. A common technique to reduce the processing burden is interrupt coalescence (IC), which groups multiple incoming packets together so that a time-stamp is generated for *each group of packets*, rather than for individual packets, so that in practice accurate timing is only available for one of the packets in the group.

Interrupt coalescence can be viewed as a sampling technique where the ideal signal, i.e., the sequence of time-stamps for individual packets, is not recorded and instead, due to coalescing, only a subset of “accurate” packet time-stamps is available (one per measurement). Various practical IC techniques have been proposed [3],

This work is supported in part by the National Science Foundation’s Networking Technology and Systems (NeTS) program, grant number CNS-0626696.

which typically use simple timers triggered by packet arrivals. The focus in these designs has been on maintaining reasonably low interrupt rates, and no formal consideration has been given to how groupings should be optimized to facilitate relevant Internet traffic analysis tasks.

The main novelty of our work is to propose for the first time a formal approach to designing IC mechanisms, so that measurements are optimized to improve Internet analysis. We are particularly concerned with quality-of-service monitoring tasks such as end-to-end link capacity estimation, available bandwidth estimation, and bottleneck detection, as well as security applications like denial-of-service (DoS) attack detection. Besides being important for proper network operation, all these tasks share a common characteristic of requiring sufficiently accurate inter-arrival timing data between packets. Certain tasks use the inter-arrival information directly, e.g., end-to-end link capacity and available bandwidth estimation [1, 4], while others use this information implicitly, e.g., DoS attack and bottleneck detection methods that use autocorrelation data obtained from the first and higher order inter-arrival histograms [2].

Based on this, the starting point for our work is that for various analysis tasks, *measurement systems should preserve both first and higher order inter-arrival statistics*. Thus, we propose that various IC approaches can be compared by obtaining first and higher-order inter-arrival statistics and quantifying, e.g., using the Kullback-Leibler divergence, how different they are from those that would be obtained from the original data, for which timing is accurate for all packets.

Our proposed optimization techniques are based on two main observations. First, existing IC methods are based on timers triggered by packet arrivals. This leads to potential biases in estimated inter-arrival times, since different packet inter-arrival patterns affect the timers, which in turn are used to generate measurements. Thus we propose measurement techniques that group packets to generate interrupts *independently of their inter-arrival times*. Specifically, before each measurement, the system decides how many packets should be aggregated into one measurement and waits as long as needed for the required number of packets to be gathered before generating an interrupt. Second, such measurement systems are completely specified by the distribution of number of packets per measurement. Thus, *we propose two different metrics to select optimal distributions of packets per measurement*. The first metric aims at maximizing the number of accurate first and higher order measurements, and is appropriate for tasks where smooth approximations of inter-arrival times are required. The second metric aims at distributing more uniformly measurements across inter-arrivals of different orders, and it is better suited for anomaly detection tasks.

To demonstrate the effectiveness of our approach in Section 4.1 we show that, for a general Poisson random process signal, as compared to standard IC methods, the optimized methods reduce the

Kullback-Leibler divergence between the ideal inter-arrival distributions and the distributions estimated using the sampled signal. Then in Section 4.2 we apply these optimized IC methods to actual Internet traffic and show we can improve the performance for a common Internet analysis application, periodic signal detection.

We build on our previous work [5], where we developed the detection method used here, following a thorough analysis of the effect of the measurement system. The key novelty of what is presented here is that rather than modify the analysis application instead we optimize the generation of measurements that work with a general class of analysis tasks.

2. PROBLEM FORMULATION

As shown in [5] a standard uniformly sampled signal representation is not appropriate for software based measurements. The packet time-stamps depend on the packet arrivals, which are irregular and unknown to the system. Thus we represent the measurement signal in vector form:

$$X(k) = \{M(k), C(k)\} \quad (1)$$

where $M(k)$ is the time-stamp of the k th group of packets measured and $C(k)$ is the number of packets in the group. This represents the actual information that has been precisely measured: each measurement provides timing information and a packet count. With this notation, an “ideal” measured signal would be $\hat{X}(k) = \{M(k), 1\}$, where exact timing is available for all packets. For a given measured signal, $X(k)$, we assume that analysis tasks will require computing histograms of first and higher order inter-arrival times, where the order refers to how many packets are received in between the measured time-stamps. If $\hat{X}(k)$ is available then the m th order inter-arrival histogram is simply obtained using $M(k) - M(k - m)$ for all k . Thus, for any order inter-arrival we would have $L - m$ measurements, where L is the total number of packets observed during the measurement period. However, if $X(k)$ produced by an IC measurement system is used, then an m -th order inter-arrival measurement can be obtained when the total number of packets recorded between two time-stamps is equal to m , that is, we can use the time difference $M(k) - M(k - j)$ for the m -th order inter-arrival histogram if we have that (note that $C(k - j)$ is not included):

$$\sum_{i=0}^{j-1} C(k - i) = m \quad (2)$$

Note therefore that when IC is used each inter-arrival histogram is based on a number of measurements smaller than $L - m$.

As stated, our goal is to optimize the coalescing of packets in a way that maximally retains the first and higher order inter-arrival information. We will use the Kullback-Leibler divergence to measure how estimates obtained using various IC methods deviate from those obtained from the ideal signal $\hat{X}(k)$. Because we wish to preserve timing information, our IC mechanism is based on selecting how many packets are aggregated into each measurement independently of their arrival times. This is in contrast with methods such as those in [3], which generates measurements when a timer counts down between packets. This biases our inter-arrival time estimates because, for example, $C(k) = 1$ only if $M(k) - M(k - 1)$ is large (in order for the timer to expire).

Let l denote the number of packets included in a measurement. We assume l is a random variable with probability mass function (pmf) $\{p(1), p(2), \dots, p(N - 1), p(N)\}$, $\sum_{l=1}^N p(l) = 1$. Thus the k -th measurement includes l packets with probability $Pr(C(k) = l) = p(l)$. Note that this pmf completely defines how the system

operates. Thus, denoting $P(N)$ the set of all possible pmfs with a maximum of N packets per measurement, our goal will be to define metrics related to the relevant analysis tasks and find the optimal pmf in $P(N)$ to optimize these metrics.

3. PROPOSED OPTIMIZATION TECHNIQUES

Define $I(n)$ to be the percentage of inter-arrival measurements retained at order n (these are the measurements of order n that can be directly derived from measured data, i.e., without using histograms derived for lower order arrivals to estimate higher order inter-arrivals). $I(n)$ is computed by taking the summation of all possible ways to get an inter-arrival of order n (as computed using (2)), this is equivalent to finding the integer partitions of n . Computing partitions is done recursively using a method such as that of Wilf, which computes the partitions in Gray Code or “minimum change” order [6]. Let $S(n)$ be the set of partitions of the integer n , and let $s_n(j) \in S(n)$ be a partition of n with j elements, i.e., $s_n(j) = \{e_1, e_2, \dots, e_{j-1}, e_j\}$ and $\sum_{i=1}^j e_i = n$. Partitions are listed in non-increasing order, without regard for order, which is undesirable for computing $I(n)$. For example, the partition $\{3, 1, 1\}$ can also be reordered as $\{1, 3, 1\}$ and $\{1, 1, 3\}$, all of which are considered different when computing $I(n)$. For a given partition, $s_n(j)$ with j elements, the number of distinct permutations is given by $\frac{j!}{j_1! \cdot j_2! \cdot \dots \cdot j_r!}$ where the j_r gives the number of like elements. In our example we have $j = 3$, $j_1 = 2$ and $j_3 = 1$ giving $\frac{3!}{2! \cdot 1!} = 3$ distinct permutations.

To compute the function $I(n)$ we define the mapping α which takes each element in the partition as the index into the pmf, then computes the product of these elements:

$$\alpha : S(n) \mapsto P(N) \\ \alpha(s_n(j)) = \alpha(\{e_1, e_2, \dots, e_j\}) = p(e_1) \cdot p(e_2) \cdot \dots \cdot p(e_j) \quad (3)$$

Finally, the function $I(n)$ is computed by:

$$I(n) = \sum_{s_n(j) \in S(n)} \frac{j!}{j_1! \cdot j_2! \cdot \dots \cdot j_r!} \alpha(s_n(j)) \quad (4)$$

For example, the only way to get an inter-arrival of order 1 is to have a measurement with one packet, thus $I(1) = p(1)$. For $I(2)$ there are two partitions 2 and 1, 1, so $I(2) = p(2) + p(1)^2$; similarly, $I(3) = p(3) + 2 \cdot p(2) \cdot p(1) + p(1)^3$.

Formulation 1 *Our first optimization seeks to maximize the total number of inter-arrival measurements of multiple orders, noting that each measurement $X(k)$ contributes to multiple inter-arrival measurements. Then our goal is:*

$$\max_{\{p(i)\} \in P(N)} \sum_{i=1}^N I(i) \\ s.t. \sum_{i=1}^N i \cdot p(i) = \hat{m}, \sum_{i=1}^N p(i) = 1 \quad (5)$$

which maximizes the percentage of inter-arrival measurements retained, up to order N , subject to the constraints that the sum of the pmf is unity and the average value is a user defined \hat{m} that gives the desired reduction in measurement rate.

The optimization is carried out in Matlab, which uses a sequential quadratic programming approach to solve the constrained, non-linear, multi-variable formulation via an algorithm adapted from [7]. This optimization results in a pmf which we characterize as ‘on/off’,

where a significant probability weight is given to $p(1)$, and the remaining weight given to a much higher order to achieve the desired average value. For example, optimizing the pmf for an average of 6 packets per measurement leads to a solution of $p(1) = .8214$ and $p(29) = .1786$. The large value of $p(1)$ helps this method achieve the maximum number of inter-arrival measurements. To illustrate, consider n consecutive single packet measurements which would produce $n-1$ first-order inter-arrivals, $n-2$ second-order, etc. Alternatively, n consecutive measurements with 2 packets produces $n-1$ second-order inter-arrivals, $n-2$ fourth-order, etc. significantly less total inter-arrivals than the previous case. However, this method suffers because the distribution of inter-arrival measurements decreases geometrically in $p(1)$. Thus, we have very few measurements available as the inter-arrival order increases. To solve this problem a first approach (see Section 4.1) is to use the inter-arrival data from the lower orders, specifically the first order data, to estimate the inter-arrival data of the higher orders.

However this leads to smooth higher order inter-arrival estimates, which miss potentially interesting anomalous events.

Thus we propose a second approach, more suited for anomaly detection, based on a different optimization metric, which seeks to distribute measurements more uniformly across orders, and thus provides good histograms at higher order based strictly on measured data (without using lower order histograms to estimate higher order ones).

Formulation 2 *Our goal is to find the pmf that produces a uniform distribution of $I(n)$ for all n .*

$$\begin{aligned} \min_{\{p(i)\} \in P(N)} & \sum_{i=1}^N (q - I(i))^2 \\ \text{s.t.} & \sum_{i=1}^N i \cdot p(i) = \hat{m}, \quad \sum_{i=1}^N p(i) = 1 \end{aligned} \quad (6)$$

which minimizes the error between a uniform distribution and the $I(n)$ functions, under the same constraints as the first optimization.

The difficulty with this problem is the selection of the parameter q , which is merely an arbitrary value that the optimization attempts to make all orders of $I(n)$ equal to. In general this optimization can be solved for any given value of q that is less than N . However, solving the problem this way only leads to the minimum error solution, i.e., $\sum (q - I(n))^2$, for that q . Instead, by relaxing the expected value condition, it is possible to solve this optimization solution for any value of $q < N$, and consequently the exact solution provides a relation between q and the expected value condition, \hat{m} .

The solution begins with $I(1)$, which assuming an exact solution exists, must be equal to q , which implies $p(1) = q$. The remaining values in the pmf can be solved for iteratively as follows:

$$\begin{aligned} I(2) = q = p(2) + p(1)^2 & \Rightarrow p(2) = (1 - q) \cdot q \\ I(3) = q = p(3) + 2 \cdot p(1) \cdot p(2) + p(1)^3 & \Rightarrow p(3) = (1 - q)^2 \cdot q \\ & \dots \\ I(n) = q & \Rightarrow p(n) = (1 - q)^{(n-1)} \cdot q \end{aligned} \quad (7)$$

The general solution, $I(n)$, can easily be recognized as the probability of success on the n th attempt given a geometric distribution. Therefore the geometric distribution, with parameter q , gives an exact solution to the second optimization formulation. Further, the expected value of the geometric distribution is $1/q$, therefore by setting $1/q = \hat{m}$ the constraints of the original formulation are met.

4. EXPERIMENTAL RESULTS

4.1. Performance Evaluation Using Synthetic Data

Since our goal was to retain as much inter-arrival information as possible following coalescence the performance metric we use is the Kullback-Leibler (KL) divergence. While not a true distance metric, KL divergence is a common measure of the difference between two probability distributions [8]. For the following analysis we simulate Internet traffic with exponentially distributed inter-arrival times. While it has been shown [9] that a Poisson process does not accurately model Internet traffic we use it here to show our solution applies to a more general class of signals, i.e., signals where packet size does not constrain minimum inter-arrival times, and show later that the method works with Internet traffic as well.

We compare the performance of our optimized IC methods, which we call OICv1 and OICv2 (where v1 represents the method derived from (5) and v2 from (6)), versus three common IC strategies: fixed pack IC (PIC), fixed time IC (TIC), and a hybrid IC (HIC) method developed by Intel [3]. The operation of PIC, TIC and HIC is described in detail in our previous paper [5].

The KL divergence is computed between the ideal signal and the signal received following IC with a measurement down-sampling rate of 6 (input/output measurement rate). For each signal we compute the inter-arrival histogram up to order 40, and measure the divergence at each order.

Figure 1(a) shows the KL divergence measured using only the data received in the input signal. The plot for PIC only contains points at multiples of the measurement down-sampling rate, for example r , because for all measurement PIC generates $C(k) = r$; thus all inter-arrivals are multiples of r . The limitation of OICv1, the on/off measurement behavior, is apparent from the figure as well. The KL divergence is very small at low orders, and for inter-arrival orders near the 'off' probability value ($p(29)$ in this example), which is caused by the combination of 'on' and 'off' measurements generating inter-arrival times with orders close to 1 and 29 in this case. While OICv1 performs well at these orders we see that the divergence increases rapidly at inter-arrival orders in between the 'on' and 'off' probability values and orders much larger than the 'off' value. Finally, we notice that, except for very low inter-arrival orders, OICv2 fares the best overall using only the data received to generate the inter-arrival histograms.

OICv1 generates the largest number of total inter-arrival measurement by generating a majority of single packet measurements. This allows OICv1 to accurately replicate the first-order histogram, which we exploit to estimate the higher-order histograms by taking the convolution of the first-order histogram multiple times. Using this modification OICv1 is able to achieve very low values of KL divergence even at high inter-arrival orders.

Figure 1(b) shows the KL divergence performance using estimates of the high order inter-arrival histograms obtained from the lower order inter-arrival data. From the figure we see that OICv1 shows the smallest KL divergence of any IC method. For some methods the KL divergence actually increases, as is the case for HIC and TIC which are not pictured because their divergence is very large.

4.2. Internet Analysis Using Actual Data

To show our optimization improves performance for actual Internet analysis applications we consider periodic signal detection. Periodic signals in Internet traffic can be caused by bottleneck network links [4] or for more insidious reasons like DoS attacks [2]. For evaluation we use the detection mechanism, periodic detection using multiple measurements (PDMM), we developed in [5], which was designed

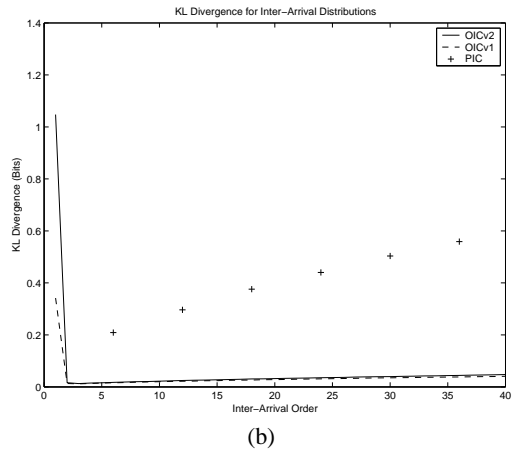
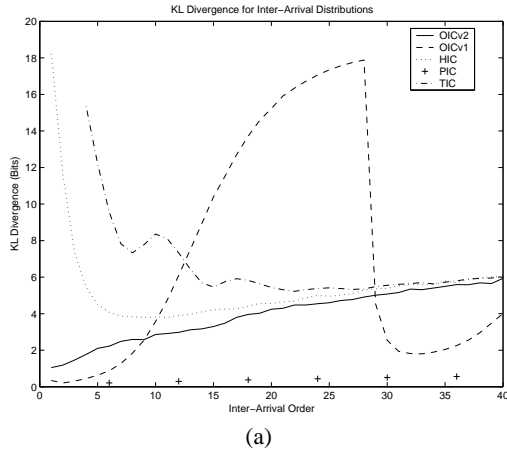


Fig. 1. KL divergence of inter-arrival distributions using (a) only received data (b) estimates from low order data

specifically to be used with software based measurements. PDMM operates by computing the higher order histograms of the measured data and uses this information to determine whether the current input traffic is random background Internet traffic or background traffic combined with a periodic signal. Because PDMM uses the higher order histogram information, accurate estimation of this information is essential for optimal performance, and therefore provides a good verification of the KL divergence results found in the previous section.

To compare the performance of our optimized IC method versus the existing techniques we perform periodic signal detection using an actual background Internet trace combined with a synthetic periodic attack signal. The background Internet trace has an average bit rate of 320 Mbps, which we merged with a 30 Mbps periodic attack signal. We compare the time to detection of the attack at a few different measurement down-sampling rates. Note that TIC is not included in the comparison because its measurements lead to an excessive rate of false positives and OICv1 is excluded for reasons explained below.

Down-sampling Rate	Coalescence Method		
	HIC	PIC	OICv2
6	1.236	4.478	.7357
7	1.299	5.310	.7594
8	1.645	6.545	.8997
10	*	8.190	.9690

Table 1. Detection performance comparing optimized IC versus standard methods. The time for HIC with down-sampling rate 10 is not included because the number of false positive detections is very high (66%).

From Table 1 we see that OICv2 performs better than the standard coalescence methods verifying the results from Section 4.1. Note that at down-sampling rate 10, OICv2 maintains detection in under one second while HIC produces more than 66% false positives (the false positive rate for all other simulations was below 10%). This demonstrates that, due to the much lower measurement rate, OICv2 could potentially be used to run detection on standard computers, as opposed to a dedicated system, or on multiple computers for a distributed detection approach.

OICv1 turns out to be unsuitable for the detection task. Using histograms generated strictly from the data produced significant false positives due to the lack of higher order inter-arrivals. Conversely, using higher order histograms estimated from the first-order inter-

arrival data no attack was detected at all because the first-order data produced smooth approximations of the higher order distributions, and the anomalous events (the periodic attack packets) do not appear in the first-order data. Thus we see that OICv1 is best suited for tasks that require significant lower-order inter-arrival data or those which demand a smooth approximation of the inter-arrival distributions. Instead, OICv2 is better for tasks where we are interested in specific events, i.e., anomalies, which are not captured by a smooth approximation.

5. CONCLUSION

By optimizing generation of measurements and removing their dependence on packet arrivals we were able to improve estimates of inter-arrival distributions, which also provided better performance in a common Internet analysis application, periodic signal detection.

6. REFERENCES

- [1] R. Prasad, M. Jain, and C. Dovrolis, "Effects of interrupt coalescence on network measurements," in *Proceeding of Passive Active Measurement (PAM) Workshop*, 2004, pp. 247–256.
- [2] U. Mitra, A. Ortega, J. Heidemann, and C. Papadopoulos, "Detecting and identifying malware: A new signal processing goal," *IEEE Signal Processing Magazine*, vol. 23, no. 5, pp. 107–111, September 2006.
- [3] Intel, "Interrupt moderation using Intel Gigabit ethernet controllers," Tech. Rep., Intel Corporation, April 2007.
- [4] D. Katabi, I. Bazzi, and Y. Xiaowei, "A passive approach for detecting shared bottlenecks," *Proceedings of the Tenth International Conference on Computer Communications and Networks*, 2001., pp. 174–181, 2001.
- [5] S. McPherson and A. Ortega, "Analysis of internet measurement systems for optimized anomaly detection system design," Tech. Rep. 0907.5233, Arxiv, 2009.
- [6] S. Pemmaraju and S. Skiena, *Computational Discrete Mathematics*, Cambridge University Press, 2003.
- [7] P.E. Gill, W. Murray, and M.H. Wright, *Practical Optimization*, Academic Press, 1981.
- [8] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, 1951.
- [9] V. Paxson and S. Floyd, "Wide-area traffic: The failure of poisson modeling," *IEEE Transactions on Networking*, vol. 61, no. 4, pp. 215–225, 1995.